Collaborative Research: Learning Discrete Mathematics and Computer Science via Primary Historical Sources

Janet Barnett, Guram Bezhanishvili, Hing Leung, Jerry Lodder,
David Pengelley, Inna Pivkina, Desh Ranjan

Jan. 2007

1 Our List of Projects

- **Summation of Numerical Powers**. The discovery of closed formulas for discrete sums of numerical powers, motivated by application to approximations for solving area and volume problems in calculus, is probably the most extensive thread in the development of discrete mathematics, spanning the period from ancient Pythagorean interest in patterns of dots to the work of Euler on a general formula for discrete summations. Initial sources from classical Greek, Indian, and Arabic traditions include Archimedes' [1] determination of the closed formula for a sum of squares, then Nichomachus, Aryabhata, and al-Karaji on sums of cubes, and al-Haytham on the sum of fourth powers. In the seventeenth century Fermat (1601–1665) [23] claimed that he could use the "figurate numbers" to solve this, "perhaps the most beautiful problem in all of arithmetic". Fermat's work was followed shortly by Pascal's [54] extensive treatise on this topic, which produced the first explicit recursive formula for sums of powers in any arithmetic progression using binomial coefficients, and which can be proved by mathematical induction. Pascal writes "I will teach how to calculate not only the sum of squares and of cubes, but also the sum of the fourth powers and those of higher powers up to infinity". This material will easily make several one-week projects at various levels, and is intended for courses in discrete mathematics and combinatorics, with PI David Pengelley as the primary author.

- **Summation of Powers, Bernoulli Numbers, and the Euler-Maclaurin Summation Formula**. Building on the previous project "Summation of Numerical Powers," David plans to author upper-level projects on summations. In the early eighteenth century, in his treatise on the beginnings of probability theory, Jakob Bernoulli (1654–1705) [6] conjectured the general pattern in the coefficients of the closed-form polynomial solution to the summation problem, writing "Indeed, a pattern can be seen in the progressions herein, which can be continued by means of this rule". Here he introduces the all-important Bernoulli numbers into mathematics; this will be explored in a project on conjecturing the general pattern, followed by a proof by mathematical induction. One or more further projects can pursue Euler's [21] development of his general 'summation formula', which provided the first proof of Bernoulli's conjecture. The Euler-Maclaurin summation formula also led to numerous other applications, including Euler's solution of the Basel problem, that $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$, which greatly contributed to his fame. Paradoxically, although Euler's summation formula almost always diverges, the method can provide arbitrarily accurate approximations to the correct answer, with Euler saying that he will sum the terms just "until it begins to diverge," and "From this it is clear that, although the series ... diverges, it nevertheless produces a true sum." These projects will thus introduce students to so-called 'asymptotic series,' an important and subtle modern applied concept. The project is designed for courses in advanced discrete mathematics, combinatorics, and possibly numerical analysis. Primary author: PI David Pengelley.

- **Logic and Truth Tables**. Modern symbolic logic is based on a system of statements which are either true or false, and then concatenated into compound statements. Beginning with Gottfried Leibniz's (1646–1716) pioneering quest for a universal symbolic language in *Dissertation on the Art*

of Combinatorics (*Dissertatio de arte combinatoria)* [26], Jerry will author a project that explores the notion of two-valued truth statements and their present-day formulation in truth tables and predicate logic. Key sources are George Boole's (1832–1916) *The Mathematical Analysis of Logic* [7] and *An Investigation of the Laws of Thought* [8], where Boole states "All the operations of language, as an instrument of reasoning, may be conducted by a system of signs . . . . And these symbols of logic are in their use subject to definite laws, partly agreeing with and partly differing from the laws of Algebra" [8]. This will be compared with Augustus De Morgan's (1808–1892) [19] use of symbols and their rules of composition, as well as informed by the influential work of Ludwig Wittgenstein (1889–1951) [67]. These pioneers' description of logic will then be contrasted with that given in modern textbooks on discrete mathematics. The project is intended for introductory courses in discrete mathematics and combinatorics. Primary Author: PI Jerry Lodder.

- **Boolean Algebra and Discrete Structures**. Following decades in which algebra involved the study of equations, the nineteenth century witnessed a fundamental shift in focus as the structures underlying algebra became more central. This new focus on structure was especially characteristic of British mathematics in the late nineteenth century. In this project, the structure of an abstract boolean algebra will be developed within the context of the work of George Boole (1832–1916) [7, 8]. Unlike the Boole project proposed by Dr. Jerry Lodder, which will focus on the notion of two-valued truth statements, this project will focus on Boolean algebras as an example of a finite discrete algebraic structure. The project will draw primarily from Boole's *An Investigation of the Laws of Thought* [7], where he attempts to "investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolical language of a Calculus, and upon this foundation to establish the science of Logic and construct its method." In other words, Boole sought to develop a purely symbolic algebra of logic. Where needed to more fully develop the modern algebraic structure of a Boolean algebra, excerpts from other works by Boole, including his *The Mathematical Analysis of Logic* [8], and those of his immediate successors in the study of logic will be incorporated into the project. These include excerpts from the work of John Venn (1834–1923) [66], Hugh MacColl (1837–1909) [49], and C.S. Peirce (1839–1914) [57]. The project is designed for courses in discrete mathematics or abstract algebra. Primary Author: PI Janet Barnett.

- **Euclid's GCD Algorithm : Recursion vs. Iteration**. The notion of recursion and recursive thinking plays an important role in computer science. For example, the divide-and-conquer paradigm for algorithm design is based critically on recursive thinking, and most problems solved via dynamic programming require recursive thinking. The notion of recursion and the use of recursive thinking for problem solving is introduced to the computer science majors very early in the standard computer science curriculum. Almost always, the notion of recursive thinking is introduced somewhat artificially, often simply announced, and without historical context. One classical example that is used to introduce recursion is Euclid's Algorithm to compute the greatest common divisor (GCD) of two integers. The so-called Euclidean algorithm is typically presented as either of the two procedures:

```
I. GCD(a,0) = 1                                  II. GCD(a,0) = 1
   GCD(a,b) = GCD (a, b mod a) if a <= b             GCD(a,b) = GCD(a,b-a) if a <= b
   GCD(a,b) = GCD(b,a) if a > b                      GCD(a,b) = GCD(b,a) if a > b
```

  Was either of these really Euclid's algorithm? Could Euclid be one of the first people to think "recursively" about solving a problem? We intend to develop a project based on Proposition 2 of Book VII of Euclid's *Elements* [20], written around 300 B.C.E., where he presents his algorithm "to find the greatest common measure of two given numbers not relatively prime" and its important corollary that "if a number measures two numbers, then it also measures their greatest common measure." The project will explore if Euclid's algorithm was recursive or iterative and how these two ways of thinking are related. It will also explore further the issues of computational efficiency of the methods resulting from these two different ways of thinking about the GCD problem. The project is designed for introductory courses in discrete mathematics and computer programming. Primary Author: Senior Researcher Desh Ranjan.

- **Induction and Recursive Thought**. A proof method that is intimately related to recursive thinking is the method of mathematical induction, often being difficult for undergraduate students to comprehend and master. Working with Dr. David Pengelley, Desh will devise projects that explore the history of "inductive proofs" and help students think of proof by induction as a natural proof method. In particular, we would like to develop integrated projects that involve use of inductive proofs to prove the correctness of recursive procedures. A historical source that seems intriguing and ripe for use in these projects is Dedekind's 1888 paper "Was sind und was sollen die Zahlen?" (translated as "The Nature and Meaning of Numbers") [18]. In this essay, where Dedekind (1831–1916) strives to "lay a foundation for the part of logic that deals with the theory of numbers," Dedekind argues for accepting the method of defining a function by induction. In the preface to the first publication of this essay , Dedekind says "As such main points I mention ... the proof that the form of argument known as complete induction (or the inference from n to $n+1$) is really conclusive and that therefore the definition by induction (or recursion) is determinate and consistent." This should bring forth the intimate connection between recursion and induction. It will also be interesting to see how recursive definitions came to play a central role in the theory of computing and in this respect Turing's and Church's original papers [65, 12] are interesting sources. Finally, Desh plans a project around the integration of recursion as an explicit construct in modern programming languages. The projects are authored for introductory to intermediate courses in discrete mathematics and computer programming. Primary Author: Senior Researcher Desh Ranjan.

- **A History of Sorting: The Emergence of Quicksort**. Sorting is the fundamental algorithmic problem in computer science, being the first step in solving many other algorithmic problems. Donald Knuth, author of *The Art of Computer Programming* [45], wrote: "I believe that virtually every important aspect of programming arises somewhere in the context of searching or sorting." Inna plans to investigate the origins of sorting algorithms, their roots in discrete mathematics, and author historical projects on sorting. The starting point is the book by Knuth, which discusses the history of sorting from the nineteenth century, when the first machines for sorting were invented. One of the projects would be on Quicksort, constructed by Tony Hoare in 1960. When Hoare became a programmer with a small computer manufacturer, his first task was to implement a sorting routine based on the best then-known algorithm (Shellsort). Hoare "greatly enjoyed the challenge of maximizing efficiency" and when he told his boss that he had invented a sorting method that would usually run faster than Shellsort, the boss bet him sixpence that he had not. Hoare won his bet. Quicksort is a comparison sort which is significantly faster in practice than other comparison sorting algorithm. Sedgewick in his paper "Implementing Quicksort programs" [62] presented "a practical study of how to implement the Quicksort sorting algorithm and its best variants on real computers". Sedgewick's paper is easy to read and can serve as an historical source for a project. The main idea of the project would be to experimentally verify results of Sedgewick by implementing the original and optimized versions of Quicksort and comparing the implementations. In an optimization, an explicit stack is used to remove the recursion. Therefore, the project will allow students to master Quicksort, practice their programming and program optimization skills, and use stack data structure. It can also be used to illustrate running time complexity of Quicksort obtained when analyzing the algorithm. The project is designed for courses in data structures, computer programming, algorithms. Primary Author: PI Inna Pivkina.

- **History of Coding and Huffman Codes**. Codes have been used since ancient times. One of the famous examples of a code is a quote from Wadsworth Longfellow's poem: "One if by land, two if by sea." This was a signal to Paul Revere on April 19, 1775, during the American Revolution about how the British would attack. Another well-known code is Morse code, invented in the mid-1830s for use in telegraphy. In Morse code, shorter codewords are used for letters such as 'e' and 't' which are more common in English. This reduces the amount of data needed for transmission which is a goal of data compression. Researchers have sought to devise an optimal (or minimum-redundancy) coding, that is, a coding which would yield the lowest possible average message length. In 1949 Claude Shannon and Robert Fano developed a systematic method to assign codewords based on probabilities of blocks [63]. However, their coding was not optimal, but only approached the optimal behavior. In 1951, Robert

Fano, a professor at MIT, assigned to his information theory students a term paper on the problem of finding the most efficient code. One of the students, David Huffman, devised an optimal method and by doing so exceeded his professor. In 1952 Huffman published a paper describing his "optimum method of coding an ensemble of messages consisting of a finite number of members" [36]. A Huffman code is a technique for compressing data allowing for savings of 20 to 90 percent, depending on the data being compressed. The data structures used in constructing these codes are priority queues and binary trees. Huffman codes are an example of prefix codes, where no codeword is a prefix of some other codeword. The proposed work includes researching the history of prefix codes, historical roots of codes in mathematics, and authorship of projects on coding algorithms. The project is designed for courses in data structures and the analysis of algorithms. Primary author: PI Inna Pivkina.

- **Networks and Spanning Trees**. Trees arise today in computer science as those graphs with the least number of edges that connect a given set of vertices. In this project the development of the idea of a *tree* is studied from the original paper of Heinz Prüfer (1896–1934) "A New Proof of a Theorem on Permutations," in 1918 [58]. Prüfer introduces the notion of a graph network to describe a net of railway connections between n-many towns in which the least possible number of connections is used, and describes several properties of this network which today have been realized as theorems about trees. Furthermore, he counts all such networks, calling the various possibilities "permutations," which in fact enumerate all labeled trees with n vertices. The project will explore Prüfer reduction of the railway problem to a combinatorial question about what today are called graphs, and investigate properties of trees identified by Prüfer. The module is designed for courses covering graph theory, trees, and minimal spanning trees. Primary Author: PI Jerry Lodder.

- **Program Correctness**. One of the greatest accomplishments in computer science has been the development of scientific principles for reasoning about program correctness [50]. Many textbooks in algorithm design discuss program correctness briefly, mostly focusing only on the concept of loop invariants. In Robert Floyd's landmark 1967 paper [24], program correctness is initially considered for flowchart programs. In this project, the students will first analyze in detail the correctness of a flowchart program taken from Floyd's paper. Next, they are asked to apply the same proof ideas to argue the correctness of an assembly program (whose structure resembles closely a flowchart program) that computes the integer square root of a number. A challenging task is to establish the correctness proof of a Quicksort implementation [25]. Projects of various difficulty levels will be written for the study of program correctness, one focusing on the basic ideas of a correctness proof, another developing flowchart and assembly programs, and a third exploring high level programming language constructs. Emphasis is on the "sets of axioms and rules of inference which can be used in proofs of the properties of computer programs" [35], and the "basis for formal definitions of the meanings of programs in appropriately defined programming languages, in such a way that a rigorous standard is established for proofs about computer programs, including proofs of correctness, equivalence, and termination" [24]. The projects are designed for advanced undergraduate courses in the analysis of algorithms. Primary Author: PI Hing Leung.

- **Arthur Cayley and Group Theory**. The seeds of group theory can be seen in several early nineteenth century mathematical developments. The common features of these apparently disparate developments were first explicitly recognized in an 1854 paper [10] by Arthur Cayley (1812–1895). As Cayley noted, "The idea of a group as applied to permutations or substitutions is due to Galois, and the introduction of it may be considered as marking an epoch in the progress of the theory of algebraical equations." Extending this notion, Cayley defined a *group* to be any (finite) system of symbols subject to certain algebraic laws, and illustrated this definition by contrasting the "system of roots of this symbolic equation $[\theta^n = 1]$" with "the system of roots of the ordinary equation $x^n - 1 = 0$". Following a demonstration of how "the distinction between [these two systems] presents itself in the very simplest case, $n = 4$," Cayley proceeded to classify all groups up to order seven. While focusing on the classification of arbitrary groups and their properties, Cayley did not neglect to motivate this abstraction through references to specific nineteenth century appearances of the group concept. These include a reference to the theory of elliptic functions, about which Cayley remarked that "systems of this form are of frequent occurrence in analysis, and it is only on account of their extreme simplicity

that they have not been expressly remarked." In this project, students will thus witness the process and power of mathematical abstraction as played out in Cayley's article. This project will also draw out several important theorems of finite group theory as students work through and elaborate on the details of Cayley's classification of groups of small order. The project is designed for courses in discrete mathematics and abstract algebra that examine the application of finite groups to coding theory. Primary Author: PI Janet Barnett.

- **Regular Languages and Finite Automata**. In 1943, McCulloch and Pitts published a pioneering treatise [51] on a model for the behavior of the nervous systems. Following up on this work, Kleene wrote the first paper in 1956 [43] on finite automata, the simplest machine model in that the machine has a finite number of possible internal states. In Kleene's historical paper [43], the finite automata model was not proposed without precedent. A substantial portion of the paper was written in the study of McCulloch-Pitts nerve nets. As stated by Kleene "we are investigating McCulloch-Pitts nerve nets only partly for their own sake as providing a simplified model of nervous activity, but also as an illustration of the general theory of automata, including robots, computing machines and the like" [43]. In the project, we follow Kleene's approach to the study of McCulloch-Pitts nerve nets as special cases of finite automata. We also study Kleene's definition of regular events (languages), and see how he reached the conclusion "By using the notion of regular events, we thus demonstrate that a McCulloch-Pitts nerve net can represent any event which any other kind of finite digital automaton can represent." The project is designed for undergraduate courses in automata theory or the theory of computation. Primary Author: PI Hing Leung.

- **Gödel's Completeness Theorem**. Kurt Gödel's (1906–1978) name is so much attached to his Incompleteness Theorems that often his other (rather numerous and important) contributions are overlooked. Among those one of the most important is Gödel's Completeness Theorem, which connects the syntactic notion of consistency with the semantic notion of satisfiability. The Completeness Theorem was the main topic of Gödel's Ph.D., and he published a version of the proof in 1930 [27], one year before his world famous Incompleteness Theorems [28]. Our modern day curriculum teaches Gödel's Completeness Theorem using a different technique, developed by Leon Henkin (1921–2006) [33, 34] two decades after Gödel's original proof. The aim of this project is to perform a comparative study of these two beautiful proofs of the Completeness Theorem, through which we hope to convey a better understanding of why every consistent set of first-order formulas has a model. We will give a careful definition of consistency and satisfiability, and show how, using only syntactic means, we can construct a model that will satisfy a given consistent set of first-order formulas. The project is designed for courses in logic and foundations of mathematics. Primary Author: PI Guram Bezhanishvili.

- **Peano Arithmetic**. The formal development of arithmetic goes all the way back to ancient Greek mathematics. However, the modern theory of arithmetic was only developed in the second half of the nineteenth century with the work of Hermann Grassmann (1809–1877), Richard Dedekind (1833–1916), and Gottlob Frege (1848–1925). But it was not until Giuseppe Peano's (1858–1932) treatise *"Arithmetices principia, nova methodo exposita"* (The principles of arithmetic, presented by a new method) [55, 56] that the axiomatic theory of arithmetic was devised as we know it today. Peano, alongside Frege and Bertrand Russell (1872–1970), is considered one of the major figures in the foundations of mathematics at the end of the nineteenth and the beginning of the twentieth century. In fact, the great Russell considers meeting Peano "a turning point of my intellectual life" [42]. It is not widely known that most of the modern set theoretic and logical notation, including the symbol $\in$ for membership relation, originated in the work of Peano. We plan to design a project on *Peano Arithmetic* based on the 1889 original historical source by Peano [55]. We will study the first-order axiom scheme of mathematical induction, define addition and multiplication recursively, and derive their main properties, such as commutativity, associativity, and distributivity, using the scheme of mathematical induction. The project is designed for courses in logic and foundations of mathematics. During the Fall Semester of 2006, Guram authored and classroom tested a first draft of this project. Primary Author: PI Guram Bezhanishvili.

- **Gödel's Incompleteness Theorems**. Kurt Gödel (1906–1978) is widely regarded as one of the most

significant logicians of all time. His work has had enormous impact on the 20th century scientific and philosophical development. Gödel's celebrated Incompleteness Theorems [28, 29] were a fatal blow to Hilbert's Program, and have forever changed the foundations of mathematics. In a recent book [30] Rebecca Goldstein calls Gödel's Incompleteness Theorems "the third leg, together with Heisenberg's uncertainty principle and Einstein's relativity, of that tripod of theoretical cataclysms that have been felt to force disturbances deep down in the foundations of the 'exact sciences.'" What are these great theorems then, and what difference do they really make? We plan to address these and related questions in the project. We will develop the technique of Gödel numbers [64], and prove that in every consistent and sufficiently expressive theory there are statements that are true, but are not provable from the axioms of the theory. Consequently, we obtain that even if a theory is rich enough to express its own consistency, it will *never* be able to prove it! The project is designed for courses in logic, foundations of mathematics, and theoretical computer science. Primary Author: PI Guram Bezhanishvili.

References

[1] Archimedes, *On Spirals, in* The Works of Archimedes, T.L. Heath (editor), Dover Publications, New York.

[2] Baldwin, D., Scragg, G., *Algorithms and Data Structures: The Science of Computing*, Charles River Media, 2004.

[3] Baldwin, D., Henderson, P., "The Importance of Mathematics to the Software Practitioner," *IEEE Software* (2002), 110–112.

[4] Baldwin, D., "Discovery Learning in Computer Science," *Proceedings of the 27th SIGCSE Symposium on Computer Science Education* (1996), 222–226.

[5] Barnett, J., Bezhanishvili, G., Leung, H., Lodder, J., Pengelley D., Ranjan, D., "Historical Projects in Discrete Mathematics and Computer Science," in *A Discrete Mathematics Resource Guide*, Hopkins, B. (editor), The Mathematical Association of America, to appear.

[6] Bernoulli, J., *Ars Conjectandi (The Art of Conjecturing)*, in *Die Werke von Jakob Bernoulli*, Naturforschende Gesellschaft in Basel, Birkhäuser Verlag, Basel, 1975, also translated by E.D. Sylla, Johns Hopkins University Press, Baltimore, 2006, and in D.J. Struik, *A Source Book in Mathematics, 1200–1800*, Cambridge, Harvard University Press, 1969; Princeton University Press, Princeton, 1986.

[7] Boole, G., *An Investigation of the Laws of Thought on which are Founded the Mathematical Theories of Logic and Probabilities*, Dover, New York, 1958.

[8] Boole, G., *The Mathematical Analysis of Logic*, Blackwell, Oxford, 1951.

[9] Calinger, R., (editor), *Classics of Mathematics*, second ed., Prentice-Hall, Engelwood Cliffs, New Jersey, 1995.

[10] Cayley, A., "On the theory of groups, as depending on the symbolic equation $\theta^n = 1$," *Philosophical Magazine*, Vol VII (1854), 40–47.

[11] Chabert, J-L., *A History of Algorithms From the Pebble to the Microchip*, Chris Weeks (translator), Springer Verlag, New York, 1994.

[12] Church, A., "An Unsolvable Problem of Elementary Number Theory," *The American Journal of Mathematics*, **58** (1936), 345–363.

[13] Cohen, M., Gaughan, E., Knoebel, A., Kurtz, D., Pengelley, D., *Student Research Projects in Calculus*, Mathematical Association of America, Washington D. C., 1992.

[14] Confrey, J., Dennis, D., "The Creation of Continuous Exponents: A Study of the Methods and Epistemology of John Wallis," *Conference Board of the Mathematical Sciences, Issues in Mathematics Education*, **6** (1996), 33–60.

[15] Davis, M., "Mathematical Logic and the Origin of Modern Computers," in *Studies in the History of Mathematics*, Mathematical Association of America, Washington D.C., 1987, 137–165.

[16] Davis, M., "First Order Logic," in *Handbook of Logic in Artifical Intelligence and Logic Programming*, vol. I, Gabbay, Hogger, Robinson (editors), Oxford University Press, Oxford, 1993, 31–65.

[17] Davis, M., *The Universal Computer: The Road from Leibniz to Turing*, W. W. Norton & Co., New York, 2000.

[18] Dedekind, R., *The Nature and Meaning of Numbers*, authorized English translation of "Was sind und was sollen die Zahlen?" by W.W. Breman, in *Essays on Theory of Numbers*, Dover, New York, 1963. (First published by Open Court publishing Company, 1901).

[19] De Morgan, A., *On the Syllogism, and Other Logical Writings*, Routledge & K. Paul, London, 1966.

[20] Euclid, *The Thirteen Books of Euclid's Elements*, T.L. Heath (editor), Dover, New York, 1956, Book VII.

[21] Euler, L., *Institutiones Calculi Differentialis (Foundations of Differential Calculus)*, in *Opera Omnia*, series I, B.G. Teubner, Leipzig and Berlin, Lausanne, 1911– .

[22] Fauvel, J., Van Maanen, J., *History in Mathematics Education*, Kluwer, Boston, 2000.

[23] Fermat, P., "Letter to Marin Mersenne, September/October 1636, and again to Roberval, November 4, 1636," in *Pierre de Fermat, Oeuvres*, P. Tannery (editor), Paris, 1891–1922, Vol. II, 70, 84–85.

[24] Floyd, R.W., "Assigning Meaning to Programs," *Proceedings of the American Mathematical Society Symposia in Applied Mathematics*, **19** (1967), 19–32.

[25] Foley, M., Hoare, C.A.R., "Proof of a Recursive Program: Quicksort," *Computer Journal*, **14** (1971), 391–395.

[26] Gerhardt, C. I., (editor) *Die Philosophischen Schriften von Leibniz*, vol. VII, Olms, Hildesheim, 1965.

[27] Gödel, K., "Die Vollständigkeit der Axiome des logischen Funktionenkalküls," *Monatsh. Math. Phys.*, **37** (1930), 349–360.

[28] Gödel, K., "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatsh. Math. Phys.*, **38** (1931), 173–198.

[29] Gödel, K., *Collected works. Vol. I*, Publications 1929–1936, Edited and with a preface by Solomon Feferman, The Clarendon Press Oxford University Press, New York, 1986.

[30] Goldstein, R., *Incompleteness: The Proof and Paradox of Kurt Gödel*, Great Discoveries, Norton & Company, New York, 2005.

[31] Grattan-Guinness, I., *The Search for Mathematical Roots, 1870–1940: Logics, Set Theories and The Foundations of Mathematics from Cantor through Russell to Gödel*, Princeton University Press, Princeton, 2000.

[32] Guzmán, M. de, "Origin and Evolution of Mathematical Theories: Implications for Mathematical Education," *Newsletter of the International Study Group on the History and Pedagogy of Mathematics*, #28, March 1993, 2–3.

[33] Henkin, L., "The completeness of the first-order functional calculus," *The Journal of Symbolic Logic*, **14** (1949), 159–166.

[34] Henkin, L., "The discovery of my completeness proofs," *The Bulletin of Symbolic Logic*, **2** (1996), 127–158.

[35] Hoare, C.A.R., "An Axiomatic Basis for Computer Programming," *Communications of the Association for Computing Machinery*, **12** (1969), 576–583.

[36] Huffman, D.A., "A method for the construction of minimum-redundancy codes," in *Proceedings of the Institute of Radio Engineers*, **40**, 9 (1952), 1098–1101.

[37] Katz, V., "An Historical Approach to Precalculus and Calculus," *Newsletter of the Humanistic Mathematics Network*, # 6, May 1991.

[38] Katz, V., "Using the History of Calculus to Teach Calculus," *Science and Education*, **2**, #3, Fall 1993.

[39] Katz, V., *A History of Mathematics: An Introduction*, second ed., Addison-Wesley, New York, 1998.

[40] Katz, V., (editor), *Using History to Teach Mathematics*, Mathematical Association of America, Washington D.C., 2000.

[41] Katz, V., (editor) *The Mathematics of Egypt, Mesopotamia, India, China, and Islam: A Sourcebook*, Princeton University Press, Princeton, New Jersey, to appear 2007.

[42] Kennedy, H.C., *Peano*, Studies in the History of Modern Science, D. Reidel Publishing Co., Dordrecht, 1980.

[43] Kleene, S.C., "Representation of events in nerve nets and finite automata," in: C. Shannon and J. McCarthy (editors), *Automata Studies*, Princeton University Press, New Jersey (1956), 3–41.

[44] Knoebel, A., Laubenbacher, R., Lodder, J., Pengelley, D., *Mathematical Masterpieces: Further Chronicles by the Explorers* Springer Verlag, New York, to appear 2007.

[45] Knuth, D.E., *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley, Reading Massachusetts, 1973.

[46] Laubenbacher, R., Pengelley, D., *Mathematical Expeditions: Chronicles by the Explorers*, Springer-Verlag, New York, 2000.

[47] Lodder, J., "Curvature in the Calculus Curriculum," *The American Mathematical Monthly*, **110**, No. 7 (2003), 593–605.

[48] Lodder, J., "Introducing Logic via Turing Machines," in *From Calculus to Computers: Using the Last 200 Year of Mathematics History in the Classroom*, Shell-Gellasch, A., Jardine, D. (editors), The Mathematical Association of America, Washington D.C., 2005, 125–133.

[49] MacColl, H., "The Calculus of Equivalent Statements and Integration Limits," *Proceedings of the London Mathematical Society* **1**, 9 (1887), 9–20.

[50] Manna, Z., "Lectures on the Logic of Computer Programming," *CBMS-NSF, Regional conference series in applied mathematics*, SIAM (1980), page 4.

[51] McCulloch, W.S., and Pitts, W., "A logical calculus of ideas immanent in nervous activity," *Bull. Math. BioPhys.*, **5** (1943), 115–133.

[52] Otero, D., "An Historical Calculus Course for Liberal Arts Students", *Newsletter of the International Study Group on the History and Pedagogy of Mathematics*, #28 (March 1993), 7–9.

[53] Pascal, B., "Treatise on the Arithmetical Triangle," in *Great Books of the Western World*, M. Adler (editor), Encyclopedia Britannica Inc., Chicago, 1991.

[54] Pascal, B., *Sommation des puissances numériques*, in *Oeuvres*, L. Brunschvieg (editor), Paris, 1908–14; Kraus Reprint, Vaduz, Liechtenstein, 1976.

[55] Peano, G., *Arithmetices principia, nova methodo exposita*, Bocca, Torino, 1889.

[56] Peano, G., *Selected works of Giuseppe Peano*, Translated from the Italian and edited, with a biographical sketch and bibliography, by Hubert C. Kennedy, University of Toronto Press, Toronto, Ont., 1973.

[57] Peirce, Ch. S., "On the Algebra of Logic," *American Journal of Mathematics 3*, 15 - 57. Also in *Writings of Charles S. Peirce - Volume 4 (1879 - 1884)*, Ch. Kloesel (editor), Indiana University Press, Bloomington and Indianapolis, (1986), 163–209.

[58] Prüfer, H., "A New Proof of a Theorem on Permutations," *Archiv der Mathematik und Physik*, 3, **27** (1918), 142–144.

[59] Pengelley, D., Pivkina, I., Ranjan, D., Villaverde, K., "A Project in Algorithms based on a Primary Historical Source about Catalan Numbers," *Proceedings of the Thirty-Seventh SIGCSE Symposium on Computer Science Education*, Vol. 37 (2006), 318–322.

[60] Rickey, F., "My Favorite Ways of Using History in Teaching Calculus," in *Learn From the Masters"*, F. Swetz, J. Fauvel, O. Bekken, B. Johansson, V. Katz (editors), Mathematical Association of America, Washington, DC, 1995, 123–134.

[61] Rickey, F., "The Necessity of History in Teaching Mathematics," in *Vita Mathematica: Historical Research and Integration with Teaching*, R. Calinger (editor), Mathematical Association of America, Washington DC, 1996, 251–256.

[62] Sedgewick, R., *Implementing Quicksort programs*, in *Communications of the Association for Computing Machinery*, **21**, n. 10 (1978), 847–857.

[63] Shannon, C.E., "A Mathematical Theory of Communication," in *Bell System Technical Journal*, **27** (July, October, 1948), 379–423 and 623–656.

[64] Smullyan, R., *Gödel's incompleteness theorems*, Oxford Logic Guides, 19, The Clarendon Press Oxford University Press, New York, 1992.

[65] Turing, A., "On Computable Numbers with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, **42** (1936), 230–265.

[66] Venn, J., *Symbolic Logic*, Macmillan, London, 1881.

[67] Wittgenstein, L., "Logisch-Philosophische Adhundlung," *Annalen der Naturphilosophie*, **14** (1921).